1 Objectif

Déploiement d'un modèle de prédiction sur des nouvelles données non étiquetées. Evaluation de l'erreur de prédiction par ré échantillonnage.

Déploiement de modèles. Le déploiement des modèles est une activité clé du Data Mining. Dans le cas de l'apprentissage supervisé, il s'agit de classer de nouveaux individus à partir des valeurs connues des variables prédictives introduites dans le modèle.

La très grande majorité des logiciels commerciaux proposent ce type d'outil. Parmi les innombrables stratégies à ce jour, le langage PMML[1] (Predictive Model Markup Language), basé sur XML, semble vouloir s'imposer. L'idée est de définir un standard reconnu de description des modèles. Ainsi, un classifieur produit par un logiciel de Data Mining quelconque pourra être exporté vers un outil de déploiement reconnaissant cette norme et intégrant un interpréteur PMML. L'implantation des modèles dans les systèmes d'information est standardisée, réduisant les coûts engendrés par le développement répétitif de solutions ad hoc pour chaque nouvelle situation à gérer.

SIPINA peut directement appliquer un arbre de décision sur un nouveau fichier non étiqueté. Petite contrainte néanmoins, le processus de déploiement doit être consécutif à l'apprentissage. Il n'est pas possible de distribuer un modèle pour l'appliquer sur de nouveaux individus en dehors de l'environnement SIPINA.

Evaluation des performances. Prédire sur des nouveaux individus, c'est bien. Mais il faut pouvoir annoncer à l'avance les performances à venir. En effet, une affectation erronée produit des conséquences négatives (ex. diagnostiquer l'absence d'une maladie chez une personne souffrante fera qu'elle ne sera pas soignée). Pouvoir évaluer la fiabilité d'un modèle prédictif est primordiale pour la décision de sa mise en production (ou non).

Habituellement, pour mesurer les performances et apprécier la structure de l'erreur, on subdivise les données en deux parties : la première, dite échantillon d'apprentissage, est utilisée pour construire le modèle prédictif ; la seconde, dite échantillon test, destinée à éprouver le modèle, est utilisée pour obtenir la matrice de confusion et les ratios de qualité. Ce schéma est recevable si les données sont abondantes. Dans le cas contraire, en fragmentant les données, il pénalise l'une ou l'autre phase du processus de modélisation. Si on réserve trop de données pour l'apprentissage, nous disposerons de trop peu de données pour l'évaluation, l'estimation des performances est peu fiable. A l'inverse, si nous cherchons à privilégier l'évaluation, la phase d'apprentissage est dégradée car sevrée des informations cruciales qu'apportent les données.

Dans le cas des petites bases, 100 observations disponibles dans notre étude, on préfèrera passer par les techniques de ré échantillonnage. Nous utiliserons la méthode

[1] http://www.dmg.org/

bootstrap dans ce didacticiel. Le but est de fournir une mesure crédible de la performance de l'arbre construit sur la totalité des données disponibles. Pour cela, on répète le schéma apprentissage-test, en constituant des données d'apprentissage de même taille que la base initiale par tirage aléatoire avec remise[2]. Les arbres individuels construits lors de ce processus ne servent qu'à l'évaluation globale, ils n'ont pas d'intérêt intrinsèque. C'est la raison pour laquelle la grande majorité des logiciels de Data Mining ne les affichent pas.

Organisation du didacticiel. Dans ce didacticiel, nous montrons comment procéder à partir d'un classeur EXCEL subdivisé en plusieurs feuilles contenant : (1) les données pour l'apprentissage du modèle, comprenant à la fois la variable à prédire et les descripteurs ; (2) les individus à classer, ne comprenant que les descripteurs, l'objectif étant d'associer à chaque individu sa classe d'appartenance.

De plus, en nous basant uniquement sur des données de la 1ère feuille (1), nous devrons annoncer les performances sur la 2nde feuille.

L'intérêt de ce didacticiel est que nous disposons en réalité des vraies étiquettes des individus dans une 3ème feuille. Cette information n'est jamais disponible en pratique. Pour nous il s'agit avant tout d'un exercice de style. On veut évaluer la précision de notre dispositif c.-à-d. vérifier si les performances annoncées sont compatibles avec les vraies performances mesurées a posteriori.

2 Données

Nous travaillons sur les données WINE[3] (wine_deployment.xls[4]). Il s'agit de classer des alcools à partir de leurs propriétés chimiques. La variable à prédire, type d'alcool, présente 3 modalités (A, B et C). La base est complétée par 12 descripteurs, tous continus.

La 1ère feuille (Data.Learning) contient les données d'étude. Nous disposons de 100 observations. Dans ces conditions, il est hors de question de les subdiviser en apprentissage et test, surtout en utilisant des arbres de décision. Nous utiliserons la totalité de l'échantillon pour la construction du modèle de prédiction. Puis par un procédé de ré échantillonnage, nous estimerons la performance du modèle en déploiement.

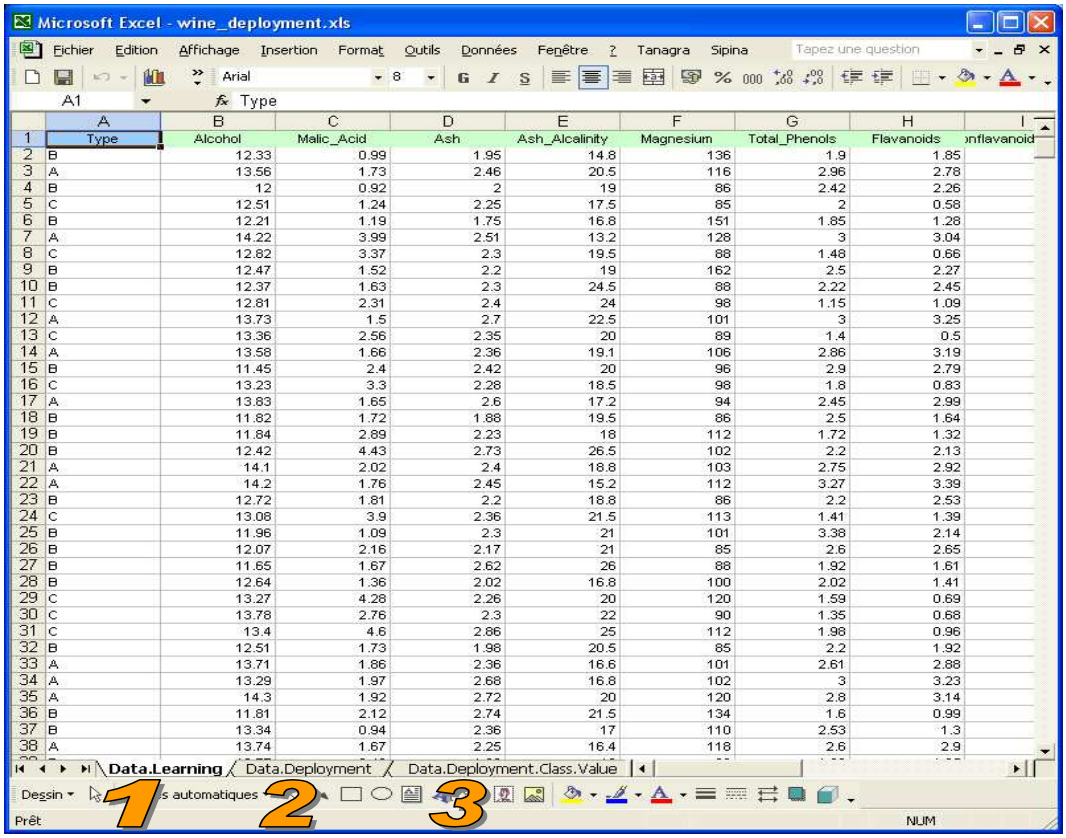
La 2nde feuille (Data.Deployment) contient les observations à classer. Seuls les descripteurs sont disponibles. Une colonne « ID », servant d'identifiant, a été ajouté pour éviter toute erreur de manipulation. Notre objectif est de compléter à cette feuille avec une colonne « prédiction » qui associe à chaque observation sa classe d'appartenance. Nous devrons alors annoncer les performances attendues, principalement la probabilité de mal classer, sur cette 2nde feuille.

[2] http://eric.univ-lyon2.fr/~ricco/cours/slides/resampling_evaluation.pdf

[3] UCI Machine Learning Repository - http://www.ics.uci.edu/~mlearn/MLSummary.html or ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine/

[4] http://eric.univ-lyon2.fr/~ricco/dataset/wine_deployment.xls

La 3ème feuille (Data.Deployment.Class.Value) contient les vraies étiquettes des individus à classer. Encore une fois, dans la réalité, ces données ne sont pas accessibles. Nous les intégrons à titre pédagogique pour apprécier la qualité de la démarche. Pour ce faire, nous copierons dans cette feuille la prédiction. Nous construirons la matrice de confusion pour calculer le taux d'erreur : cela permettra de contrôler la crédibilité de l'erreur annoncée dans la phase précédente.



	Type	Alcohol	Malic_Acid	Ash	Ash_Alcalinity	Magnesium	Total_Phenols	Flavanoids	onflavanoid
2	B	12.33	0.99	1.95	14.8	136	1.9	1.85	
3	A	13.56	1.73	2.46	20.5	116	2.96	2.78	
4	B	12	0.92	2	19	86	2.42	2.26	
5	C	12.51	1.24	2.25	17.5	85	2	0.58	
6	B	12.21	1.19	1.75	16.8	151	1.85	1.28	
7	A	14.22	3.99	2.51	13.2	128	3	3.04	
8	C	12.82	3.37	2.3	19.5	88	1.48	0.66	
9	B	12.47	1.52	2.2	19	162	2.5	2.27	
10	B	12.37	1.63	2.3	24.5	88	2.22	2.45	
11	C	12.81	2.31	2.4	24	98	1.15	1.09	
12	A	13.73	1.5	2.7	22.5	101	3	3.25	
13	C	13.36	2.56	2.35	20	89	1.4	0.5	
14	A	13.58	1.66	2.36	19.1	106	2.86	3.19	
15	B	11.45	2.4	2.42	20	96	2.9	2.79	
16	C	13.23	3.3	2.28	18.5	98	1.8	0.83	
17	A	13.83	1.65	2.6	17.2	94	2.45	2.99	
18	B	11.82	1.72	1.88	19.5	86	2.5	1.64	
19	B	11.84	2.89	2.23	18	112	1.72	1.32	
20	B	12.42	4.43	2.73	26.5	102	2.2	2.13	
21	A	14.1	2.02	2.4	18.8	103	2.75	2.92	
22	A	14.2	1.76	2.45	15.2	112	3.27	3.39	
23	B	12.72	1.81	2.2	18.8	86	2.2	2.53	
24	C	13.08	3.9	2.36	21.5	113	1.41	1.39	
25	B	11.96	1.09	2.3	21	101	3.38	2.14	
26	B	12.07	2.16	2.17	21	85	2.6	2.65	
27	B	11.65	1.67	2.62	26	88	1.92	1.61	
28	B	12.64	1.36	2.02	16.8	100	2.02	1.41	
29	C	13.27	4.28	2.26	20	120	1.59	0.69	
30	C	13.78	2.76	2.3	22	90	1.35	0.68	
31	C	13.4	4.6	2.86	25	112	1.98	0.96	
32	B	12.51	1.73	1.98	20.5	85	2.2	1.92	
33	A	13.71	1.86	2.36	16.6	101	2.61	2.88	
34	A	13.29	1.97	2.68	16.8	102	3	3.23	
35	A	14.3	1.92	2.72	20	120	2.8	3.14	
36	B	11.81	2.12	2.74	21.5	134	1.6	0.99	
37	B	13.34	0.94	2.36	17	110	2.53	1.3	
38	A	13.74	1.67	2.25	16.4	118	2.6	2.9	

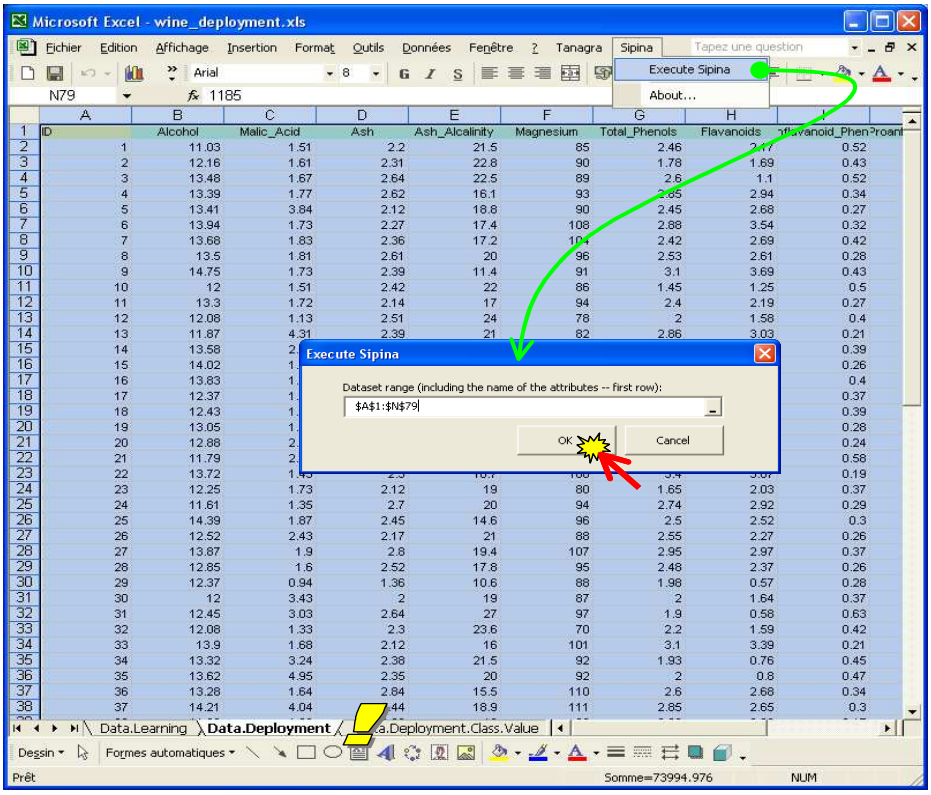
3 Apprentissage et déploiement

3.1 Préparation des données de déploiement

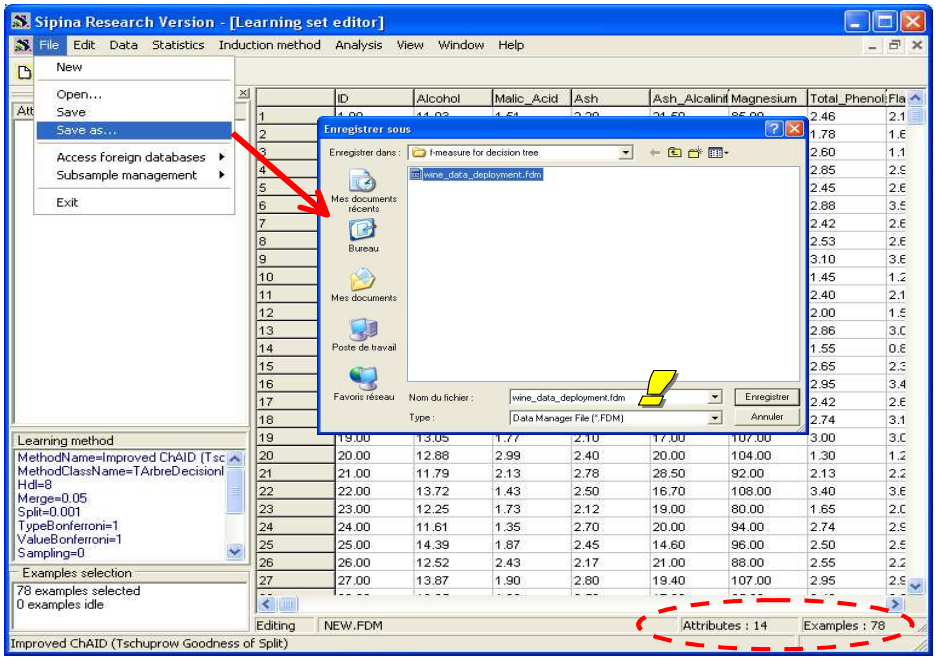
Il est possible de lancer directement le logiciel SIPINA à partir d'EXCEL via la macro complémentaire qui accompagne le logiciel[5]. Cependant, s'agissant du déploiement sur un nouveau fichier, nous sommes obligés de sauvegarder le fichier des individus à classer dans un format propre à SIPINA (*.fdm).

Nous sélectionnons les données dans la feuille « Data.Deployment ». Nous activons le menu SIPINA / EXCEUTE SIPINA. Nous vérifions que la plage de données est correctement sélectionnée, puis nous validons.

[5] Voir les didacticiels en ligne : http://eric.univ-lyon2.fr/~ricco/doc/sipina_xla_installation.htm pour l'intégration de la macro complémentaire dans EXCEL ; http://eric.univ-lyon2.fr/~ricco/doc/sipina_xla_processing.htm, pour son utilisation.



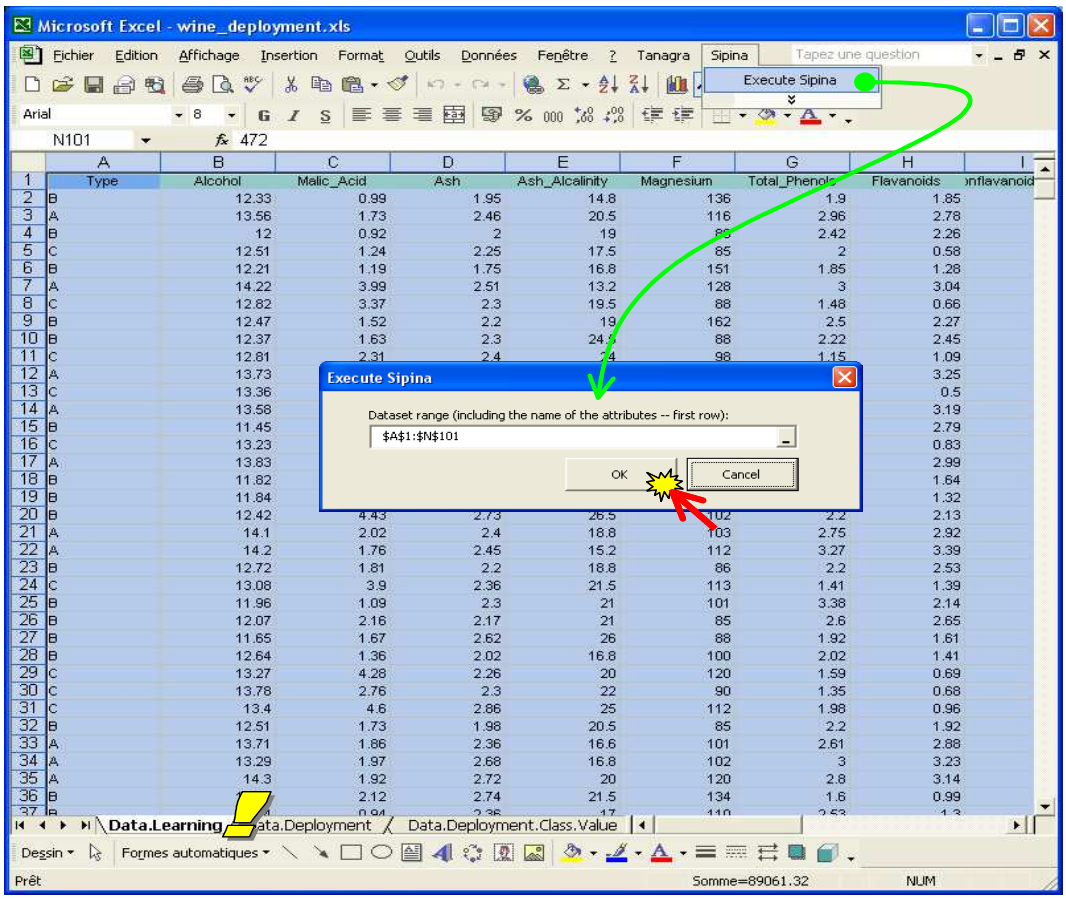
SIPINA est automatiquement démarré, les données chargées via le presse-papier. Le logiciel a bien reconnu 14 colonnes et 78 observations. Dans cette première phase, nous nous bornerons à sauvegarder les données au format binaire « .fdm ». Nous activons le menu FILE / SAVE AS, et dans la boîte de dialogue qui apparaît nous spécifions le nom du fichier, « wine_data_deployment.fdm » par exemple.



Nous pouvons fermer SIPINA et revenir dans le classeur EXCEL maintenant.

3.2 Chargement des données d'apprentissage dans SIPINA

Pour construire notre arbre de décsion, nous sélectionnons les donnés de la feuille « Data.Learning ». De nouveau, nous activons le menu SIPINA / EXECUTE SIPINA. Après avoir vérifié la sélection de données, nous validons.



SIPINA est automatiquement démarré et les données chargées.

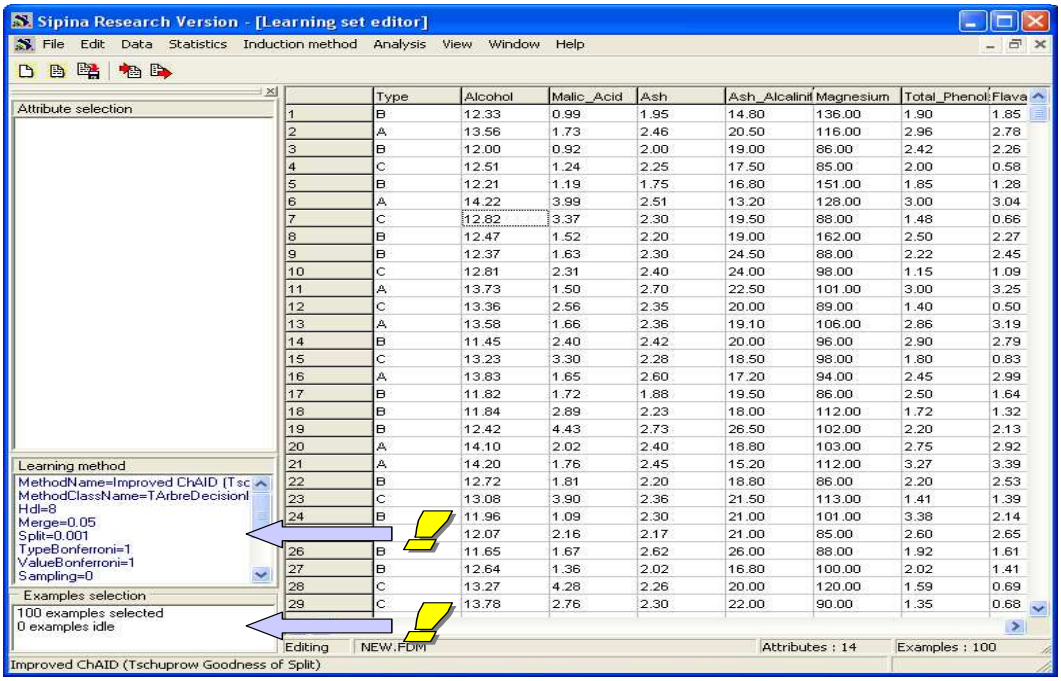
Il y a 100 observations dans cette partie des données. SIPINA les sélectionne automatiquement pour l'induction.

3.3 Construction d'un arbre de décision

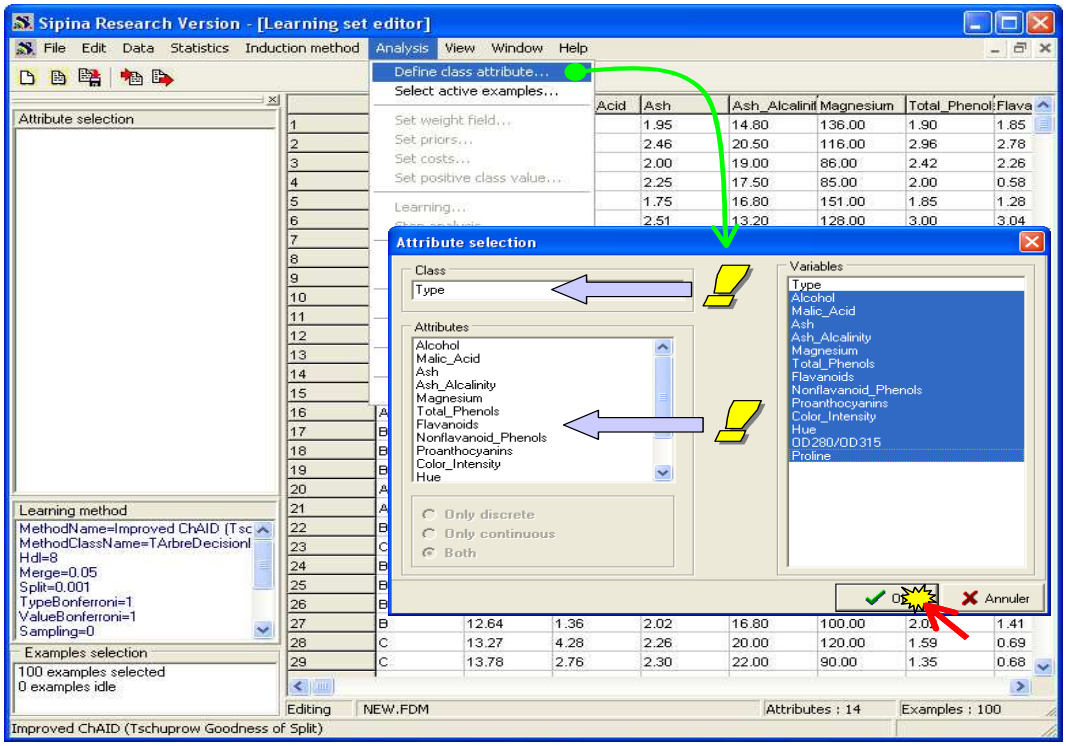
SIPINA opte pour la méthode IMPROVED CHAID lors du démarrage du logiciel. Cette variante de l'algorithme CHAID (Kass, 1980) est très rapide. Elle convient très bien pour une première appréhension d'un problème[6]. Il est bien entendu possible de choisir un autre algorithme si cela nous semble judicieux (voir le menu INDUCTION METHOD / STANDARD ALGORITHM pour obtenir la liste des techniques implémentées). Pour l'heure, nous nous en tenons à la méthode par défaut.

[6] Voir http://eric.univ-lyon2.fr/~ricco/cours/slides/arbres_decision_cart_chaid_c45.pdf pour mieux appréhender ce qui différencient les principales méthodes d'induction des arbres de décision.

Concernant les observations, SIPINA les sélectionne toutes pour l'apprentissage[7]. Ces informations sont résumées dans la partie gauche de la fenêtre principale.

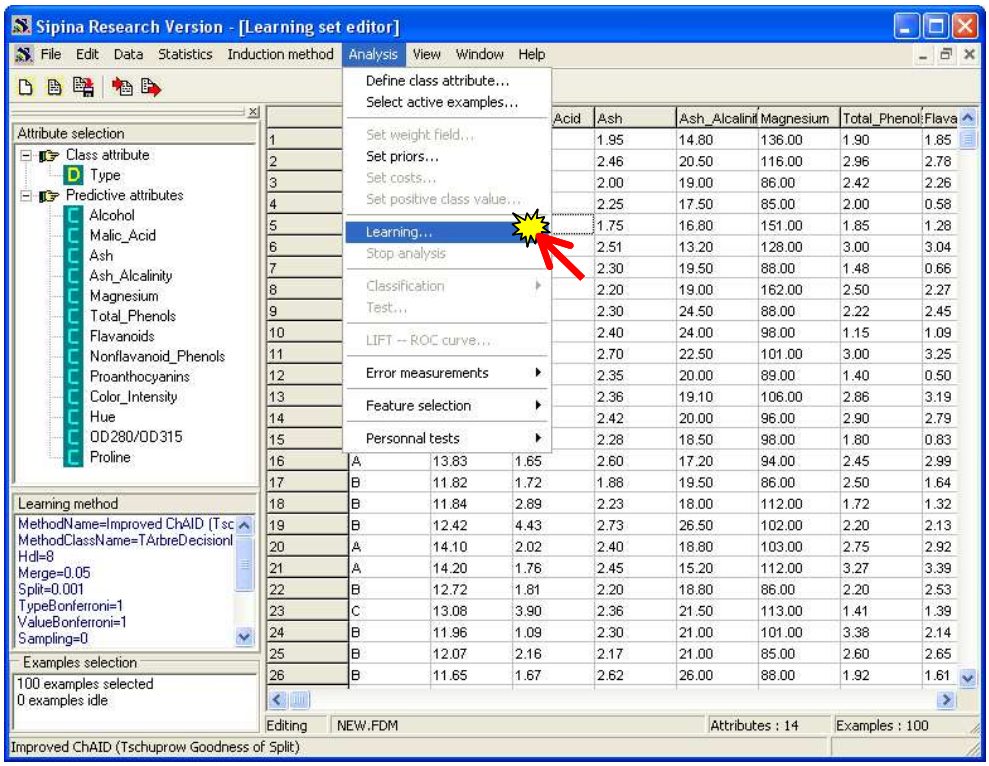


Nous devons maintenant indiquer à SIPINA la variable à prédire et les variables prédictives. Pour ce faire, nous activons le menu ANALYSIS / DEFINE CLASS ATTRIBUTE. Une boîte de dialogue apparaît, nous plaçons en CLASS (TARGET) la variable TYPE, en ATTRIIBUTES (INPUT) les variables allant de ALCOHOL à PROLINE.

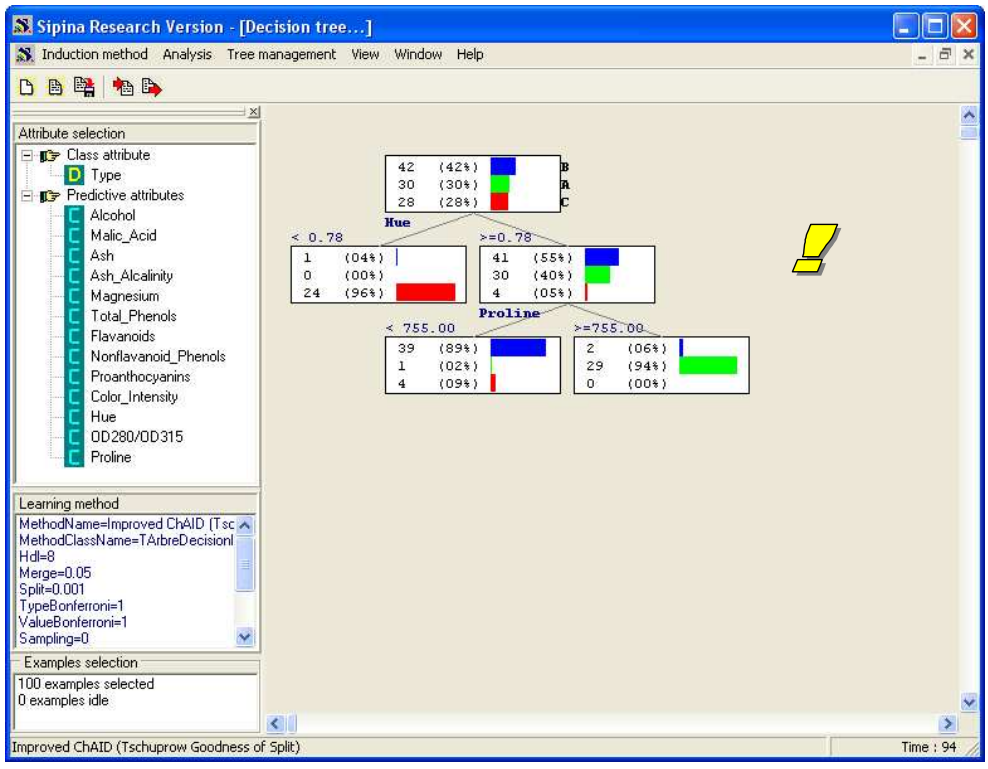


[7] Ce paramètre peut être modifié si nous voulons réserver une partie des données pour les tests.

La sélection est résumée dans la partie gauche de la fenêtre. Nous pouvons maintenant lancer l'apprentissage. Nous cliquons sur le menu ANALYSIS / LEARNING.



L'arbre de décision s'affiche, il est relativement simple dans notre exemple.

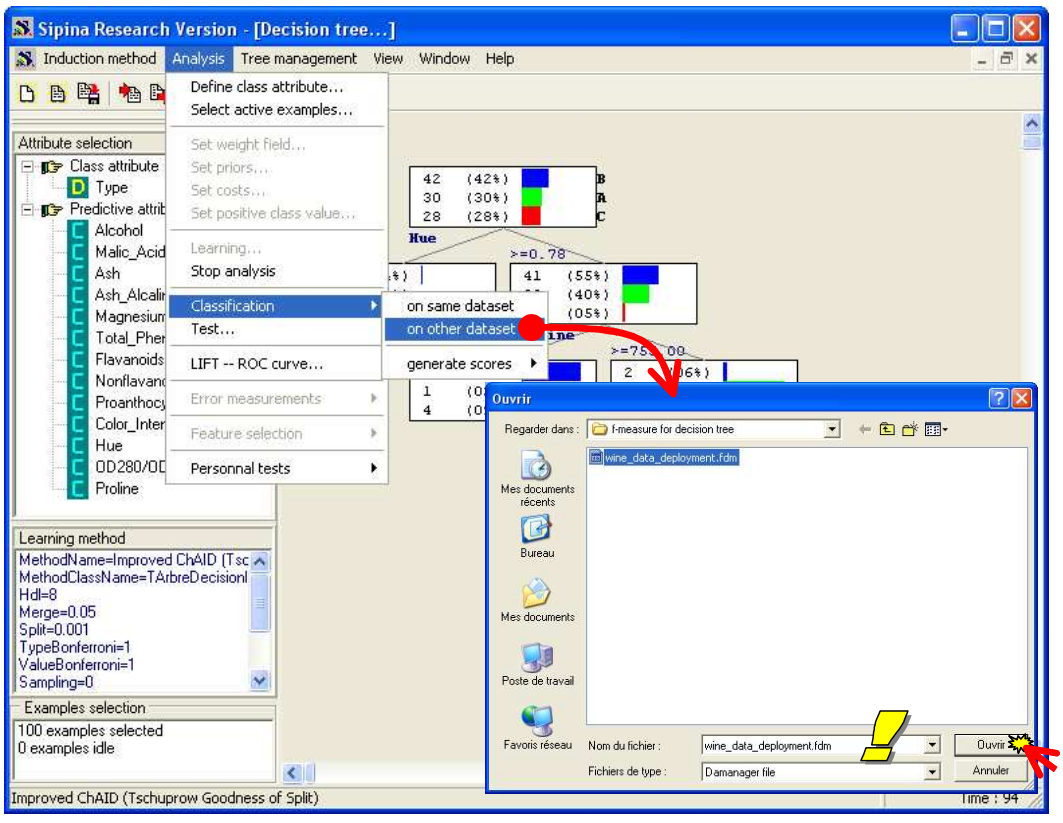


3.4 Déploiement sur le second fichier

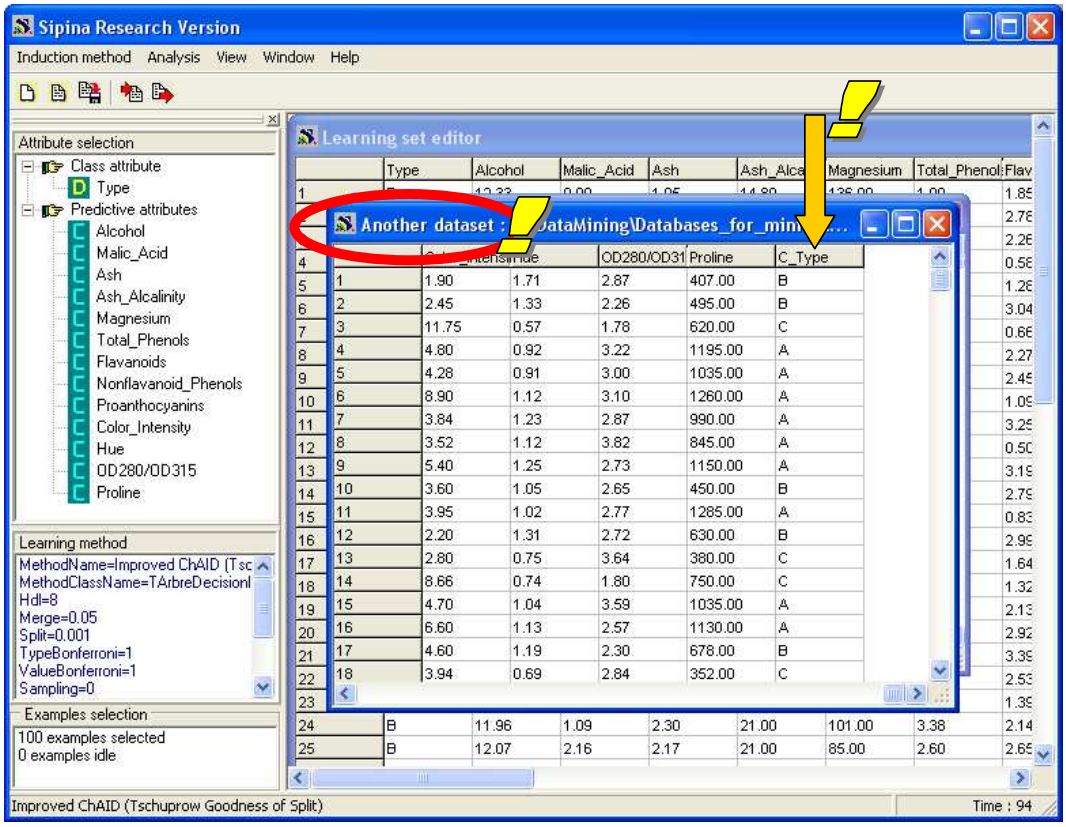
Arrive l'étape délicate. Nous voulons appliquer cet arbre sur de nouvelles données. Nous souhaitons prédire le groupe d'appartenance de chaque individu.

SIPINA peut réaliser cette opération pour les fichiers de type *.FDM, d'où la conversion effectuée initialement. Autre point important, SIPINA se contente des variables présentent dans l'arbre pour la prédiction. Il n'est donc pas nécessaire que la structure du fichier de déploiement soit identique au fichier d'apprentissage (nombre de variables, position des variables). Heureusement. En revanche, il est absolument obligatoire que les noms de variables qui apparaissent dans l'arbre soient rigoureusement identiques aux noms de variables correspondantes dans le fichier à classer. La procédure est sensible à la casse.

Pour exécuter le déploiement, il nous faut sélectionner le menu ANALYSIS / CLASSIFICATION / ON **OTHER DATASET**. Une boîte de dialogue apparaît, nous devons sélectionner le fichier préalablement converti « WINE_DATA_DEPLOYMENT.FDM ».



Deux opérations se succèdent : (1) le fichier est chargé dans une nouvelle fenêtre de données ; (2) une colonne est ajoutée dans la grille, elle reprend le nom de la variable à prédire en le préfixant par « C_ », dans notre cas, le nom de variable est « C_TYPE ».

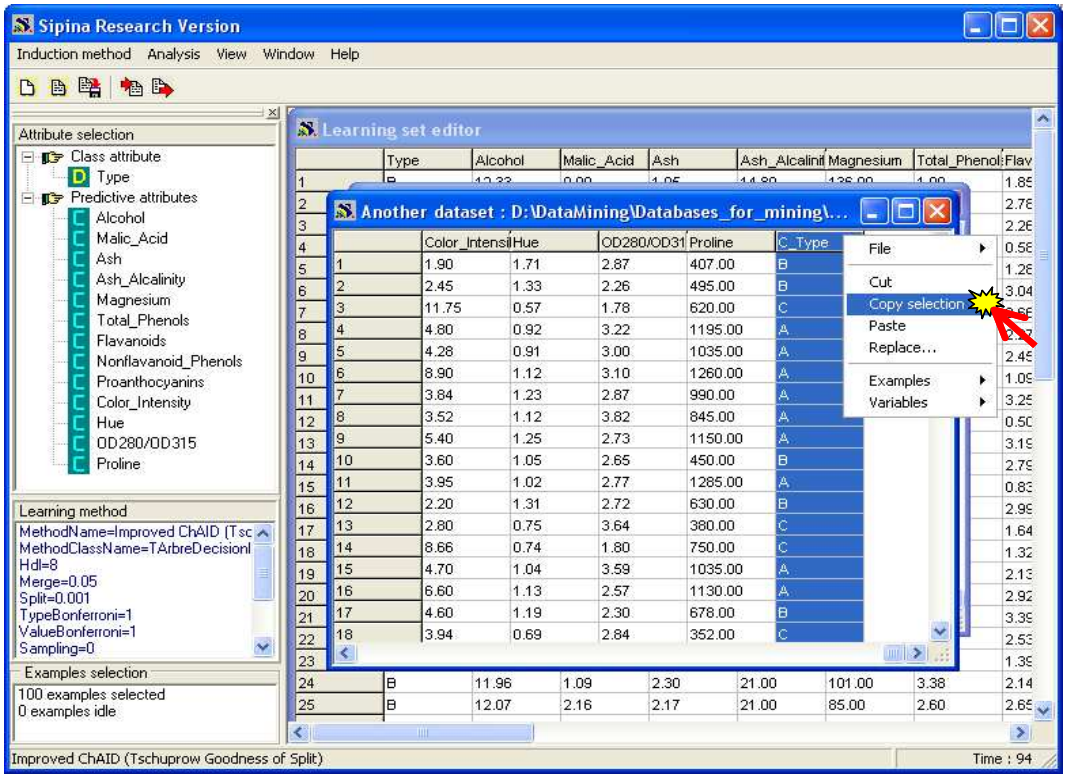


Cette nouvelle variable est définie exactement de la même manière que la variable à prédire. Elle est catégorielle avec les mêmes modalités. Dans notre cas, le premier individu se voit attribué la catégorie « B », le second « B » également, le troisième « C », etc.

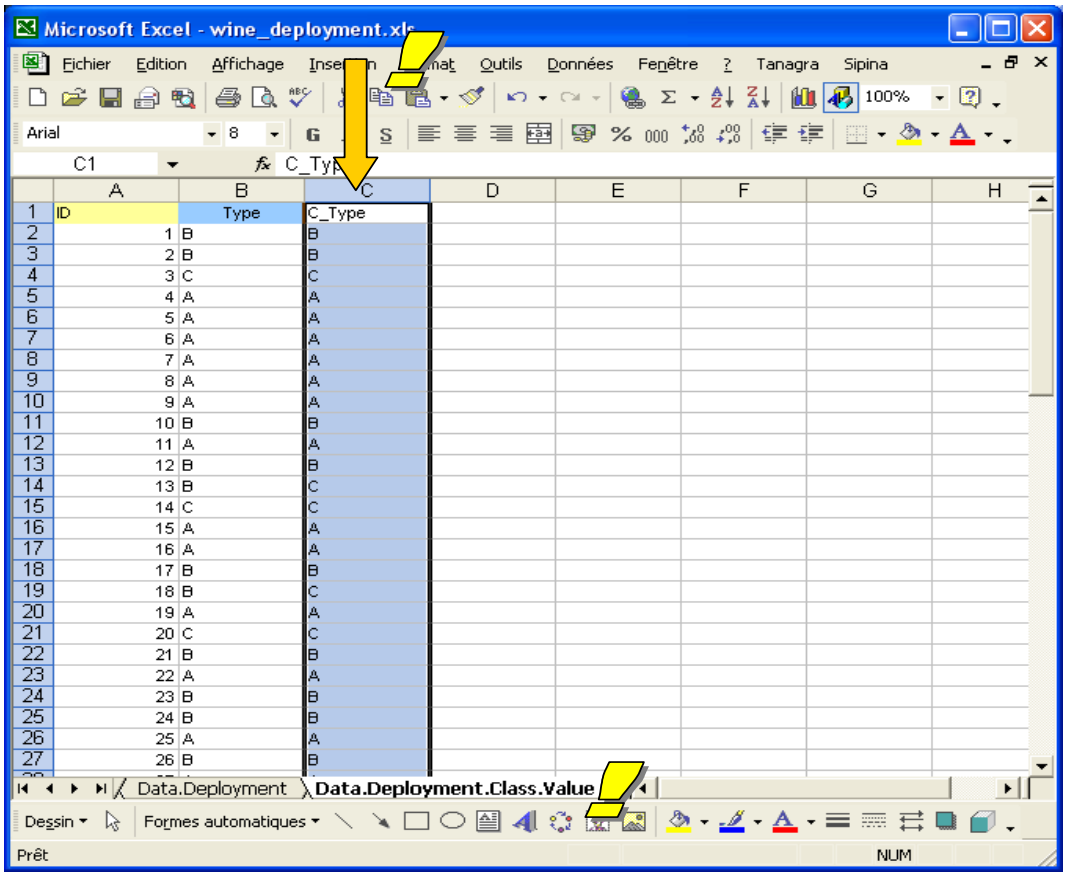
Remarque : Nous aurions pu effectuer nous même le classement dans un tableur en créant une colonne « prédiction », et en appliquant les règles d'affectation avec la fonction « =SI(...) » d'EXCEL. C'est un exercice que je préconise souvent pour que l'on se rende bien compte du mécanisme de la prédiction. Bien sûr, dès que les règles commencent à être complexes, nous sommes bien content que SIPINA, ou tout autre logiciel de Data Mining digne de ce nom, puisse le faire facilement.

3.5 Récupération de l'étiquette des individus

Afin de préparer la suite des évènements, nous allons conserver les étiquettes prédites dans notre classeur EXCEL. Un copier coller convient. Il faut pour cela cliquer sur l'en-tête de la colonne nouvelle créée, d'activer le menu contextuel (clic droit de la souris), puis de copier la sélection avec COPY SELECTION.



Nous revenons dans EXCEL. Le plus simple pour nous est d'activer la feuille contenant les vraies valeurs de TYPE pour les individus à classer (DATA.DEPLOYMENT.CLASS.VALUE), nous collons les données en C1.



Laissons là ces informations pour l'instant. Nous les exploiterons plus tard. Rappelons que nous sommes dans une configuration particulière, nous connaissons en réalité les vraies étiquettes des individus à classer.

4 Evaluation des performances

Tout déploiement doit s'accompagner d'un indicateur indiquant la fiabilité de la projection. En effet, le décideur a besoin de mesurer les risques qu'il encourt lorsqu'il affecte une étiquette à un nouvel individu. Le taux d'erreur est un indicateur privilégié dans la majorité des cas. Il est d'interprétation simple, il estime la probabilité de nous tromper lors du classement.

Comment le calculer dans notre cas ? L'idéal aurait été de disposer d'un fichier étiqueté supplémentaire qui nous sert de données test. Ce n'est pas les cas ici.

Autre approche possible, nous aurions pu subdiviser au préalable les données en apprentissage et test, réaliser la construction de l'arbre sur la première partie, puis en évaluer les performances sur la seconde. C'est une bonne démarche, celle qui faut privilégier dans la pratique. On dispose ainsi d'une estimation fiable des performances... pour peu qu'il y ait suffisamment d'individus dans l'ensemble test.

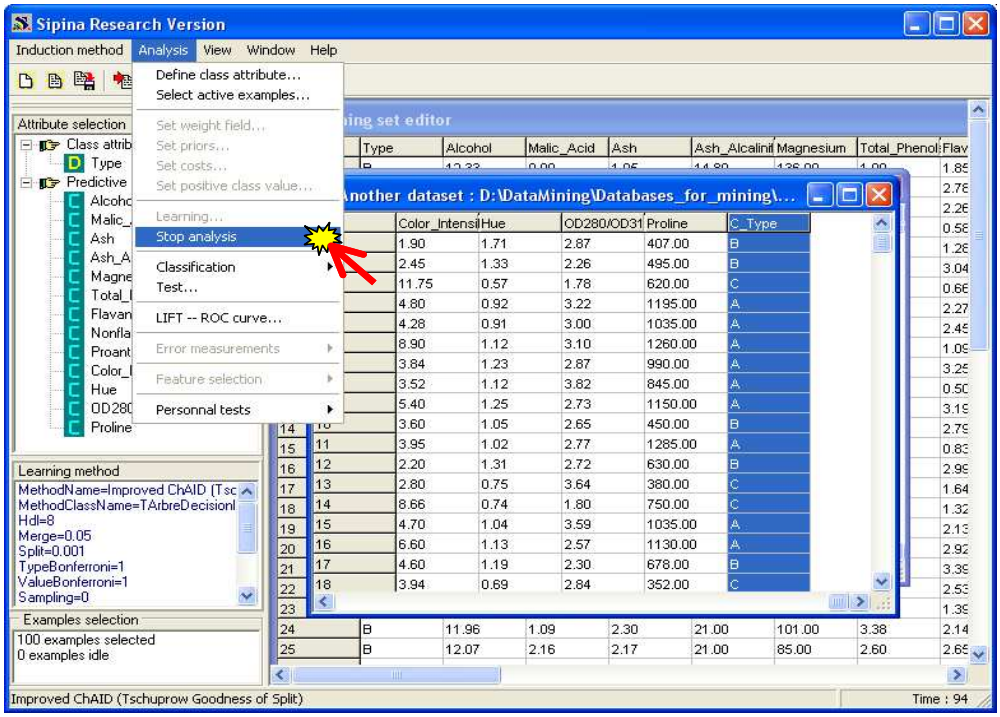
Malheureusement, dans notre exemple, cette approche est totalement inappropriée. L'effectif disponible est déjà très faible, 100 observations, la subdiviser pénaliserait la construction du modèle sans que l'on dispose de suffisamment d'observations pour en évaluer les performances. C'est d'autant plus vrai que nous mettons en œuvre les arbres de décision, réputés très instables, et de ce fait gourmands en observations.

Dans notre configuration, nous devons nous tourner vers les méthodes de ré échantillonnage. Ces méthodes, naturelles dans les publications scientifiques où on se bat rageusement à coups de dixièmes de pourcents d'erreur, sont peu comprises dans les études sur données réelles. Il s'agit ni plus ni moins que d'une technique d'estimation du taux d'erreur du modèle élaboré sur l'ensemble des données. Visualiser les modèles construits à chaque étape du processus n'a aucun intérêt. De même, il n'est pas possible d'effectuer ce type de traitement si l'on veut intervenir interactivement dans la construction de l'arbre. La procédure ne tient que si nous sommes à biais d'apprentissage égal (méthode + paramétrage).

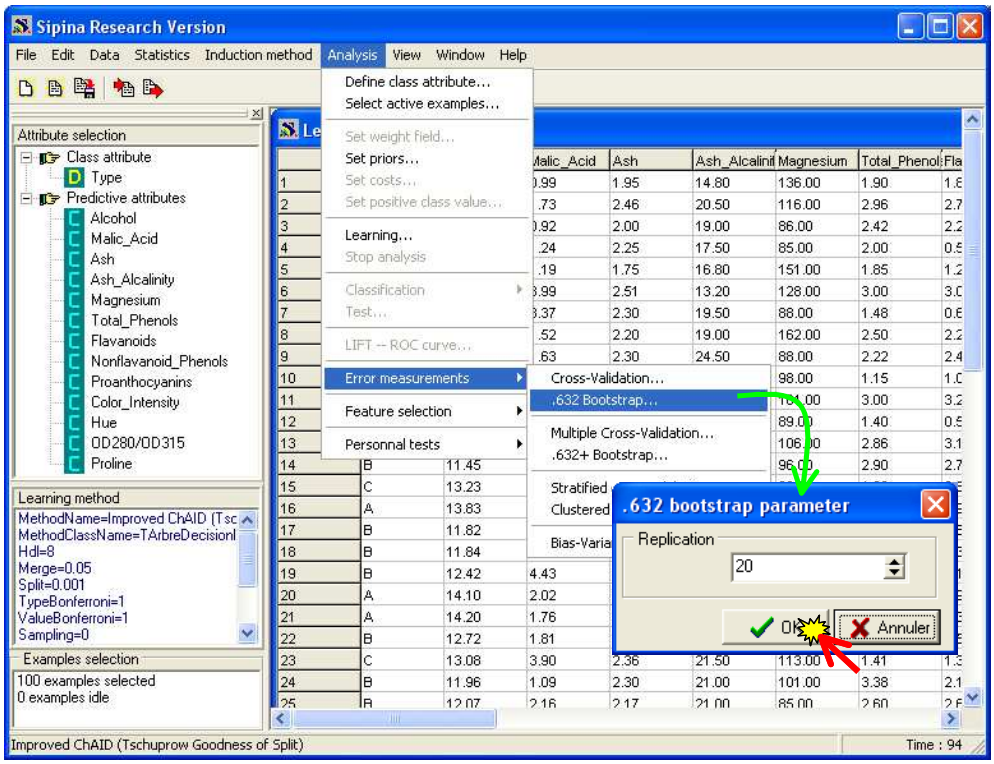
4.1 Bootstrap

La validation croisée et le bootstrap sont les deux techniques de ré échantillonnage les plus utilisées. On préfèrera le bootstrap dans notre cas.

Pour lancer la procédure, nous devons tout d'abord stopper l'analyse courante. Nous sélectionnons le menu ANALYSIS / STOP ANALYSIS, ce qui a pour conséquence de fermer les fenêtres intermédiaires, y compris celle de l'arbre.



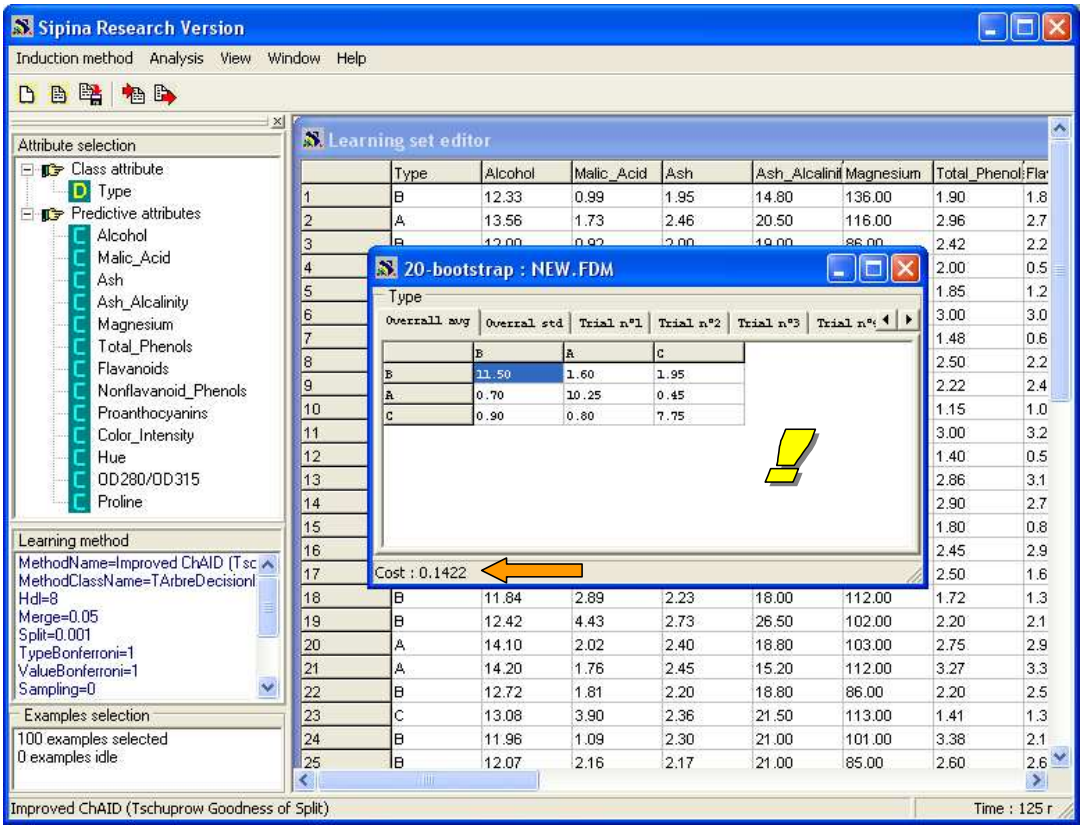
Puis, nous sélectionnons le menu ANALYSIS / ERROR MEASUREMENTS / .632 BOOTSTRAP. Une boîte de dialogue apparaît, SIPINA propose de réitérer 20 fois le processus Apprentissage / Test[8]. C'est amplement suffisant dans la plupart des situations. Nous validons en cliquant sur OK.



[8] Voir http://eric.univ-lyon2.fr/~ricco/cours/slides/resampling_evaluation.pdf ; et http://bioinformatics.oxfordjournals.org/cgi/content/full/21/15/3301

Selon la rapidité de la méthode d'apprentissage et la taille du fichier, la procédure est plus ou moins longue. Pour nous, c'est quasiment instantané. Les arbres sont réputés rapides, et nous manipulons un petit fichier.

Une matrice de confusion résumant toutes les étapes intermédiaires est affichée. Il est possible de visualiser le détail des résultats (Trial N°x), mais cela a peu d'intérêt pratique.



Dans notre cas, le taux d'erreur mesuré en validation croisée est de 0.1422. Cela veut dire que la probabilité de mal classer un individu lors du déploiement est de 14.22%. Par ailleurs, la structure de la matrice de confusion n'appelle pas de commentaires particuliers. Il n'y a pas une mauvaise affectation qui apparaît (largement) plus fréquemment que les autres.

4.2 Vérification sur les données de déploiement

L'intérêt de notre exercice est que nous disposons en réalité des étiquettes des individus sur le fichier de déploiement, nous pouvons vérifier si l'erreur estimée en ré échantillonnage est crédible ou pas.

Encore une fois, cela n'est pas possible dans les études sur données réelles, nous devons nous fier à la performance annoncée par la technique de validation que nous avons choisie. De la qualité de la démarche mise en place dépendra la crédibilité du taux d'erreur que nous proposons. Par exemple, si nous nous acharnons à manipuler les paramètres de la méthode pour trouver le modèle plus efficace en bootstrap, il est parier

que l'erreur annoncée dans ce cas tendra à être optimiste, et reflètera d'autant moins les performances lors du déploiement.

Revenons dans notre classeur EXCEL. Nous actions la feuille DATA.DEPLOYMENT.CLASS.VALUE, nous cherchons à mesurer les vraies performances sur le fichier de déploiement. Le plus simple est de réaliser un tableau croisé dynamique[9], un tableau de contingence qui croise TYPE en colonne B, et C_TYPE en colonne C.

Nombre de Type	C_Type			
Type	A	B	C	Total
A	28	1		29
B	2	23	4	29
C		4	16	20
Total	30	28	20	78

Le taux d'erreur est $\varepsilon = \frac{1+2+4+4}{78} = 14.10\%$

Quelle est la moralité de tout ceci ?

Nous constatons que l'erreur annoncée par la technique de ré échantillonnage est pour le moins précise. Elle est de 14.22%. C'est même assez idyllique à vrai dire. Même si les techniques de ré échantillonnage sont performantes, nous atteignons rarement ce niveau d'exactitude dans les conditions courantes. Il faut surtout y voir une aptitude de ces méthodes, le bootstrap ou la validation croisée, à produire une assez bonne estimation de l'erreur. Remarquons également que nous sommes dans une situation favorable, la très faible complexité de l'arbre nous prémunit de tout sur ajustement sur les données, condition propice pour une estimation consistante de l'erreur.

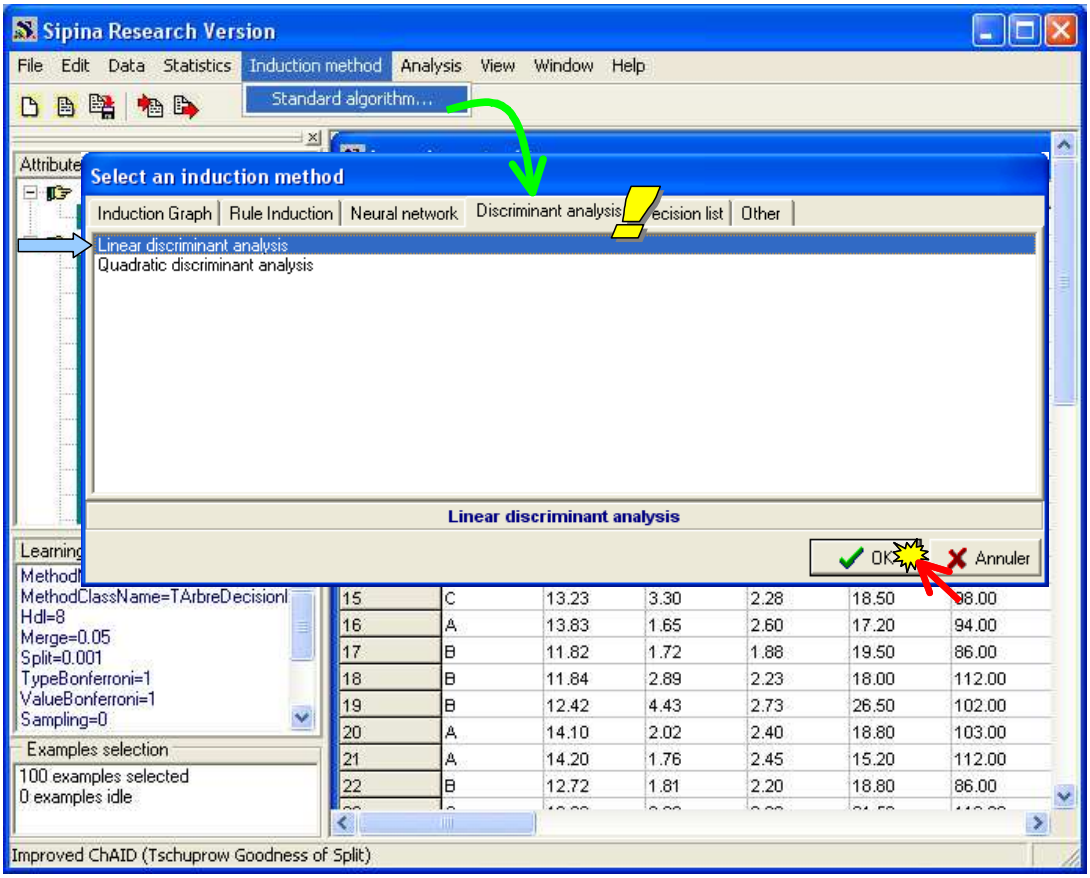
5 Concernant les autres méthodes supervisées

SIPINA est essentiellement connu pour les arbres de décision. En réalité, sa bibliothèque de méthodes, toutes supervisées, est assez étendue. Dans le cadre du déploiement, la procédure est totalement harmonisée. Si nous optons pour une autre méthode, il suffit de la sélectionner, puis de ré itérer les différentes étapes décrites ci-dessus.

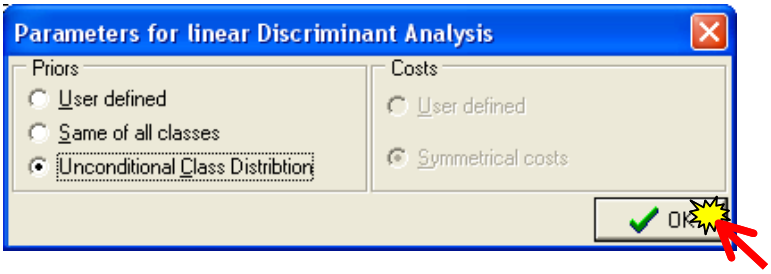
Prenons l'exemple de l'analyse discriminante. Nous activons le menu INDUCTION METHOD / STANDARD ALGORITHM. Une boîte de dialogue apparaît, nous cliquons sur l'onglet DISCRIMINANT ANALYSIS, puis nous sélectionnons la LINEAR DISCRIMINANT ANALYSIS[10].

[9] Pour l'élaboration d'un tableau croisé dynamique sous EXCEL, se reporter à d'excellents documents pédagogiques accessibles en ligne tels que http://www.lecompagnon.info/excel/tableaucroise.htm ou http://cherbe.free.fr/XL_avance.html#TCD

[10] Voir http://fr.wikipedia.org/wiki/Analyse_discriminante_linéaire pour la description de la méthode.



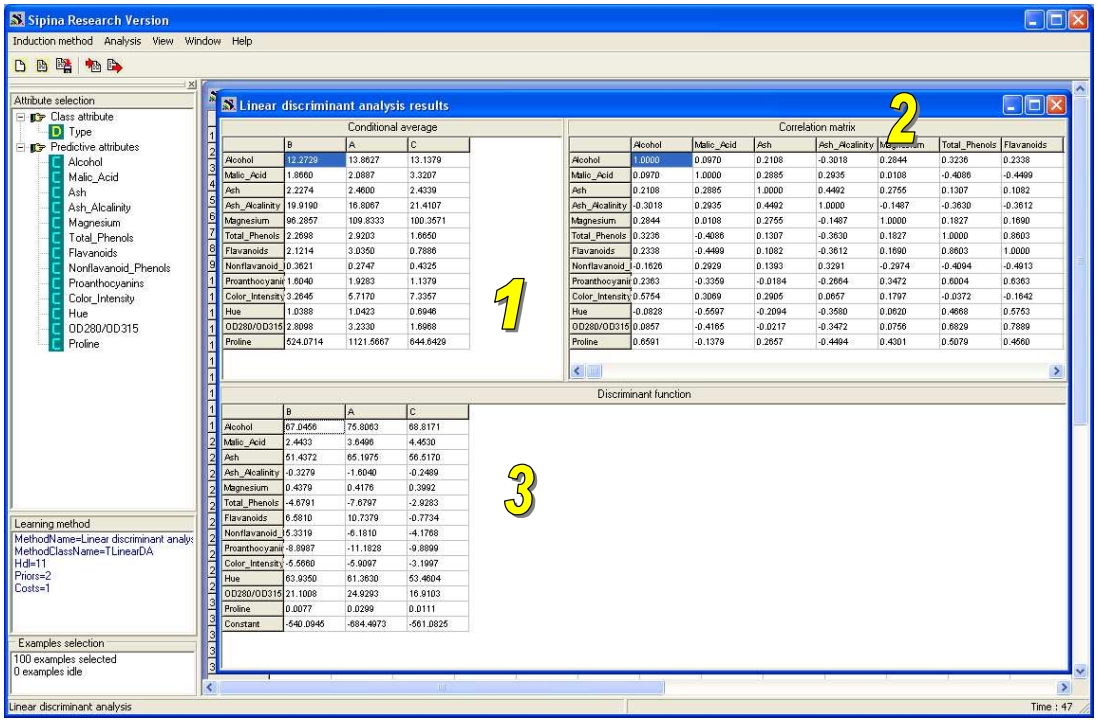
Une boîte de dialogue apparaît, elle nous sert à paramétrer la méthode. Dans notre cas, nous nous contentons de valider les propositions par défaut.



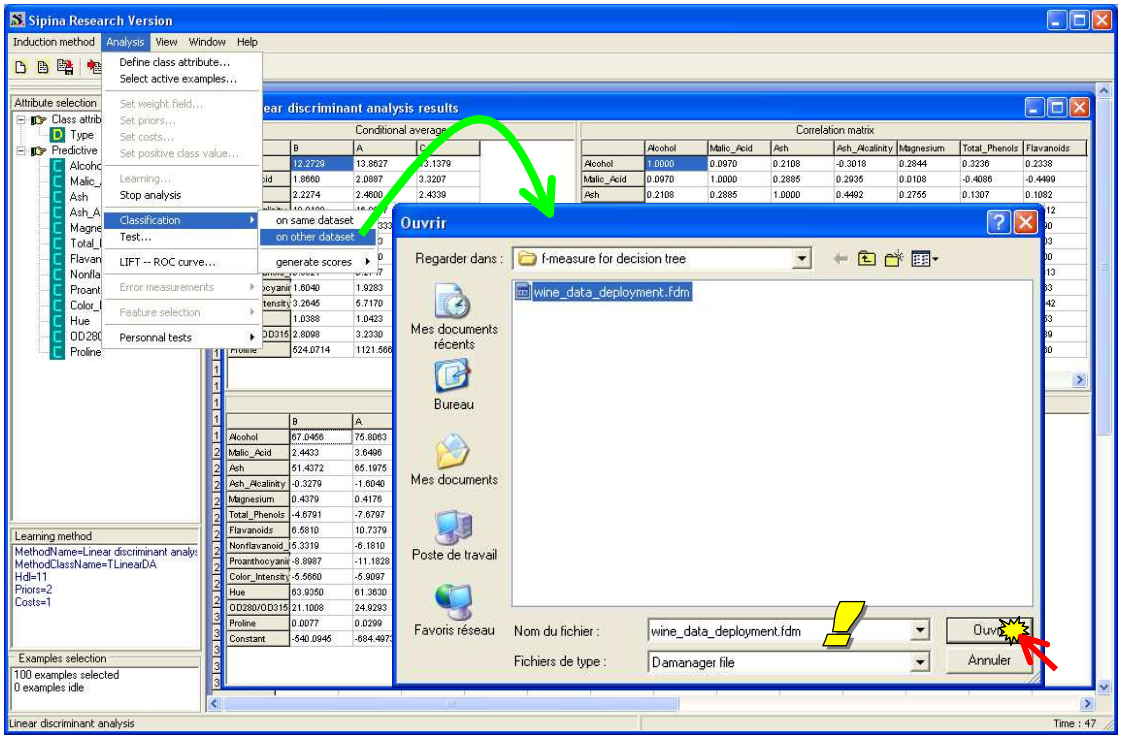
Il ne nous reste plus qu'à relancer l'analyse via le menu ANALYSIS / LEARNING.

Une nouvelle fenêtre de résultats apparaît[11], elle comporte : (1) les moyennes des descripteurs conditionnellement aux valeurs de la variable à prédire, ces informations donnent une idée (très partielle, on ne tient pas comte des covariances) sur leur rôle dans la discrimination ; (2) la matrice de corrélation, pour identifier les variables redondantes ; (3) les fonctions de classement, qui permettent de classer les individus.

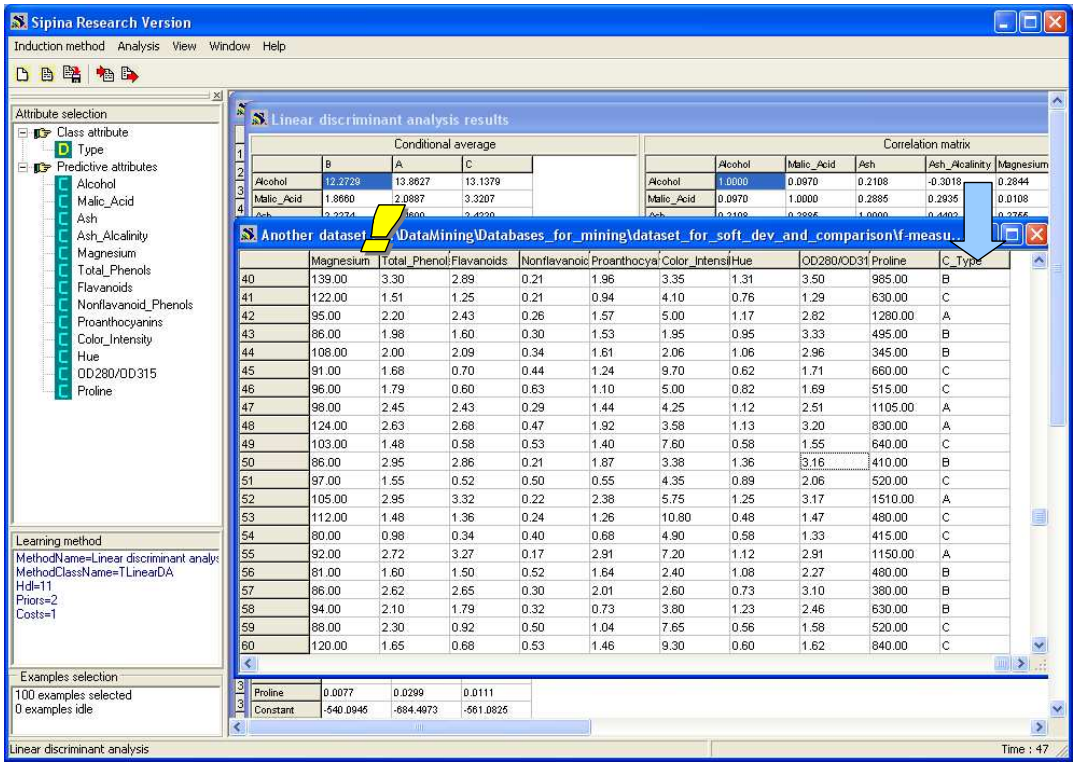
[11] On a l'essentiel, certes. Mais je reconnais que la présentation est un peu confuse, limitée par la disposition sous forme de grille. Cela fait partie des éléments que l'on a cherché à améliorer dans TANAGRA. Cette section du tutoriel sur l'analyse discriminante nous sert uniquement à illustrer l'aspect générique de la démarche.



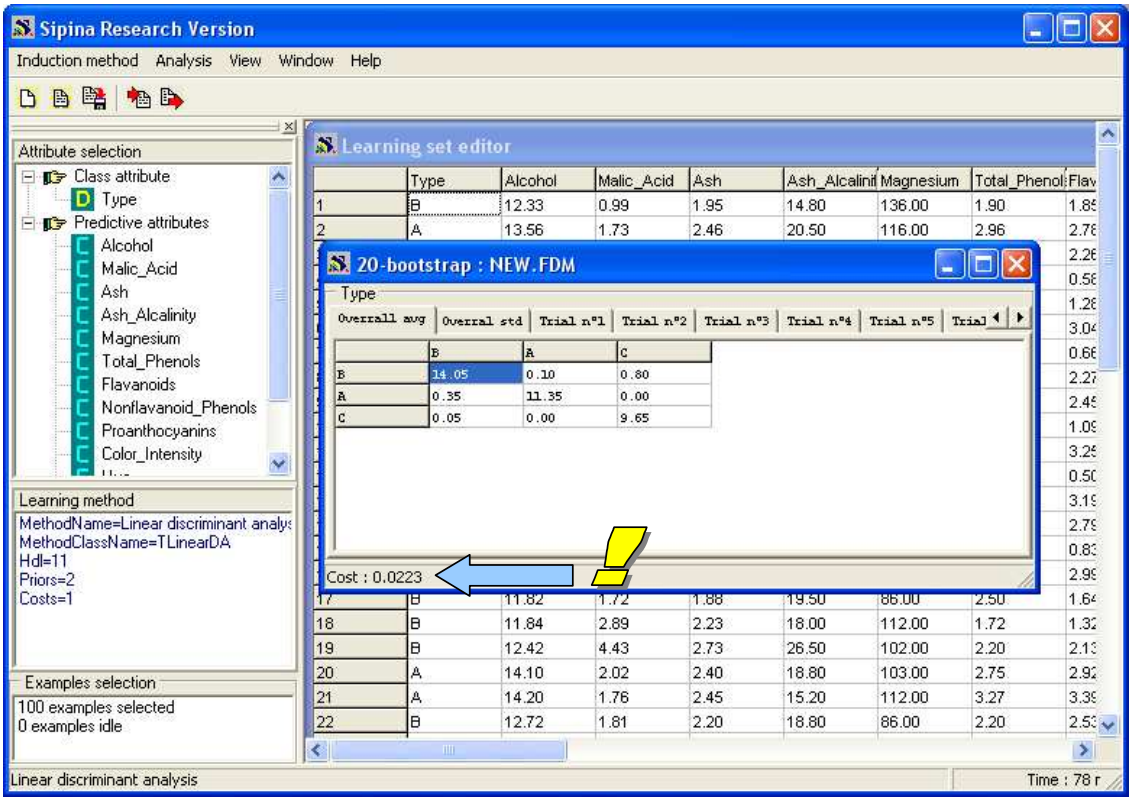
Il ne nous reste plus qu'à appliquer ce modèle prédictif sur les données de déploiement, le même que précédemment. La démarche est la même : activer le menu ANALYSIS / CLASSIFICATION / ON OTHER DATASET. Dans la boîte de sélection qui vient, on choisit le fichier « wine_data_deployment.fdm ».



La grille apparaît, comportant la description des individus et la colonne additionnelle correspondant à la prédiction de l'analyse discriminante.



De nouveau, nous pouvons évaluer le taux d'erreur par bootstrap. Nous interrompons la session d'apprentissage (ANALYSIS / STOP ANALYSIS). Puis nous lançons l'évaluation de l'erreur par bootstrap (ANALYSIS / ERROR MEASUREMENTS / .632 BOOTSTRAP).



Dans le cas de l'analyse discriminante, sur ce fichier, il est de 2.23%. Tout à fait remarquable si nous le comparons avec le taux annoncé par les arbres de décision. Finalement, il est plus judicieux d'utiliser l'analyse discriminante si on veut prédire avec la meilleure précision possible le type des alcools à partir de leurs caractéristiques chimiques.

6 Conclusion

Dans ce didacticiel, nous montrons comment appliquer un modèle prédictif construit avec SIPINA sur un nouveau fichier de données non étiqueté.

Cette tâche, dite de déploiement, très courante, doit pouvoir être mise en œuvre facilement après la phase de modélisation. On doit éviter à tout prix les manipulations hasardeuses en jonglant avec les fichiers et les règles de prédiction à appliquer.

Autre spécification importante, il ne doit pas être nécessaire que le fichier à étiqueter soit de la même structure que le fichier de données initial. Le logiciel doit être capable de retrouver, parmi les variables disponibles, celles qui sont prises en compte dans le modèle de prédiction. SIPINA effectue cette opération en cherchant simplement les correspondances entre les noms de variables.

La solution proposée par SIPINA a le mérite de la simplicité. Mais c'est avant tout un outil pédagogique. Il présente les forces et les faiblesses de son cahier des charges. Dans un contexte professionnel, il apparaît indispensable que l'on puisse sauvegarder le modèle à déployer afin de l'appliquer sur un nouveau fichier, indépendamment du logiciel de Data Mining, dans un système d'information entre autres. On peut citer la diffusion des règles dans des formats standardisés tels que le SQL, si l'on s'en tient aux arbres de décision. Plus généralement, on imagine à terme que lorsque le PMML aura atteint un bon niveau de maturité et de notoriété, son aspect générique devra décupler les possibilités de diffusion des modèles prédictifs.